Week 1 Notes

Lecture 1 & 2

1 Supervised Learning (lecture 1)

Def: *model*; a model is some structure that uses parameters to perform a function. An example of a model is $y = Wx + b$, which is a linear model. Technically speaking, we should write a linear model as (refer to 1.1 for more info):

$$\hat{y} = \sum_{j=1}^{m} w_j \cdot X_j + b = xW^T + b$$

The intuition is that, the parameters (i.e. W and b) can be changed based off the data so that the model performs the correct function/prediction/output. Note that X is the input and the output is y. Also note that this assumes there is a linear relationship between the input and output, which is not the case all the time. Therefore, a model can be any polynomial or non-polynomial function. We can think of a machine learning model as a mathematical function, F, with 1 or more inputs likeso:

$$\hat{y} = F(X_1, ..., X_m)$$

For example, the inputs of a weather model can be the humidity, temperature, air speed, etc. And the output could be whether it will rain (regression model) or a probability distributon of what the weather will be like (classification model). In a regression model the output $y \in \mathbb{R}$ while the output of a classification model is a probability distribution. A classification model with 2 classes is a binary classification model, a model with more than 2 classes is a multi-class classificatiton model.

Def: *supervised learning*; when we can train a model with labeled data. For example, given an image of a dog the label will be "dog". Formally, we can express this as:

$$D = \{(x_1, y_1), ..., (x_n, y_n)\}$$

where D represents our dataset, x_i is the input vector to the i^{th} sample and y_i is the corresponding label to the i^{th} sample. Note that, we call x_i the input vector because we can only pass in numbers. For example, an RGB image would be a vector of size $(H \times W \times 3)$. Similarly, the output has to be a vector and cannot be something like the label "dog". We need encode the label using a technique such as *one-hot encoding* for categorical data or Universal Sentence Encoder (USE) for sentences in NLP. Ultimately, our goal in supervised learning is to learn a function F such that for a new pair of data (x, y), we have $F(x) \approx y$ with high probability.

1.1 Linear Models and Why $\hat{y} = xW^T + b$?

In actuality, W, x, b, and $\hat{y}$ are all matrices.
The typical linear (fully connected) layer in torch uses input features of shape (N,∗,in_features) and weights of shape (out_features,in_features) to produce an output of shape (N,∗,out_features). Here N is the batch size, and ∗ is any number of other dimensions (may be none).
Computing derivative for this is easier than

2 Regression (lecture 1)

2.1 Linear Regression

If we have a linear model (i.e. $y = Wx + b$) finding the particular parameters (W and b) for this linear function is called linear regression. We can think of linear regression as the line of best fit.

3 Loss Function (lecture 2)

4 Gradient Descent (lecture 2)